

BAKER BOTTS L.L.P.  
30 ROCKEFELLER PLAZA  
NEW YORK, NEW YORK 10112

---

TO ALL WHOM IT MAY CONCERN:

Be it known that WE, ULRICH BOELKENS, GERHARD HEINEMANN, GEORG STEINLEIN, and ALEXANDER WAGENPFEIL, citizens of Germany, whose post office addresses are Mittlere Kreuzgasse 13, 90403 Nurnberg, Germany; Leipziger Str. 20A, 91058 Erlangen, Germany; Hammerstatt 7, 95448 Bayreuth, Germany; and Wallweg 9A, 91341 Roettenbach; Germany, respectively, have invented an improvement in

OPEN DRIVE REGULATOR, AND A METHOD FOR  
OBTAINING SOFTWARE FOR AN OPEN DRIVE REGULATOR

of which the following is a

SPECIFICATION

FIELD OF THE INVENTION

[0001] The invention relates to an open drive regulator and a method for software generation for an open drive regulator. The expression drive regulator means, for example, converter devices and their software for operation of electrical and/or hydraulic actuators (for example motors).

BACKGROUND OF THE INVENTION

[0002] What are referred to as intelligent drives for central and decentralized automation are known from the prior art. In this case, various components in a system

carry out the tasks of open and closed-loop process control in a hierarchical structure. For example, a servo converter can signal the corresponding control data on a direct path to a control system. If there are a number of regulators in one station, they are connected to one another via a communication bus, which ensures direct data matching.

[0003] Intelligent drives are also used for specific open and closed-loop control tasks, for example for printing and winding technology. For this purpose, an intelligent drive provides functions which are matched to the requirements of the application by means of control software. The intelligent drive provides a library of different open and closed-loop control elements for these application-specific requirements. These are conventional modules from general control and automation technology, process regulators, technology regulators, monitoring/diagnosis algorithms, and start-up transmitters.

[0004] The SIMODRIVE drive regulator from Siemens AG is known in the prior art, particularly from the functional description "SIMODRIVE 611 digital, SIMUMERIK 840D/810D", Order No.: 6SN1197-OAA80-OAP6, issue 10.2000. These regulators contain drive functions such as closed-loop control of the four-quadrant circuit including limits for synchronous and asynchronous motors with/without rotation, speed/position detection, rotation speed control, control messages/alarm reactions, and diagnosis functions.

[0005] The systems known from the prior art to this extent have the disadvantage that they are not what are referred to as open systems. Openness is a software function which is now in general use in the field of control engineering. It offers the user an

efficient capability to integrate his own specific solutions in the overall system beyond the basic functionality that is preset in the factory. However, there are no such open software functions for the field of drives. In fact, software packets with functionalities that are defined and collated in the factory are offered in this field. Since these software systems are not open systems, the management, servicing and maintenance of customer-specific variants involve a large amount of management effort (software production, software handling, software marketing).

[0006] The object of the present invention is to provide an open drive regulator and an improved method for software generation for an open drive regulator.

#### SUMMARY OF THE INVENTION

[0007] The present invention allows software generation on the basis of function objects. These are compiled individually and can be preprocessed in the form of library routines. In a further step, the software for the open drive regulator is generated from the individually compiled function objects and from the library routines. This concept can be implemented generally on the central unit and on the intelligent peripheral components. Furthermore, jump-in points for customer-specific upgrades are offered in the function objects.

[0008] One particular advantage of the invention is that OEM customers are provided with the capability of including their own applications (functions or diagnoses) additively to the basic system. Links are provided for this purpose to the control

infrastructure (for example parameter descriptions, messages, warnings, alarms, function call lists, file functions, documentation production, links to the control hardware).

[0009] A further particular advantage of the invention is that existing function elements of the basic system can be omitted and/or replaced by customer-specific function elements. The infrastructure of the omitted function elements is in this case not transferred to the generated software. This has the advantage of performance improvement, in particular with regard to the program running time and resource conservation, particularly with regard to the amount of memory required.

[0010] An additional particular advantage of the invention is the capability of loading function objects on line from any desired sources (internal memory media, external memory media (as examples: CD, Internet)) via the existing communication buses, and the capability of integrating them in the program sequence.

[0011] Furthermore, the present invention allows real-time applications to be provided, which are supported by a control capability, which can be tailor-made on a customer-specific basis, within a start-up tool.

[0012] The capabilities of an open drive allow fast, flexible solutions tailor-made for the customer, and also shorten the time-to-market for the basic system, since the development team no longer has to develop customer-specific solutions. "Special developments" on the basis of a validated basic structure are economic both for the user and for the manufacturer.

[0013] A further advantage of the present invention is that the capability of the customer to develop his own OEM software variants allows him to protect his know-how,

and to keep it secret. For this purpose, an OEM customer can use appropriate development tools, procedures and documents provided by the manufacturer, so that he can carry out his own further development of the basic functionality. An advantageous feature in this case is the alignment of the basic software architecture into functional units with standardized interfaces, which can be combined with one another in a defined manner on the basis of a basic functionality. In addition to algorithms in different time slices, a function object and/or an instantiated function include:

- parameters
- control messages, warnings, alarms
- initialization routines
- test scripts
- start-up options associated with the function.

[0014] The invention is not dependent on the chosen programming languages; however, a preferred software architecture is that provided by using a C++ environment.

### DRAWINGS

[0015] A preferred exemplary embodiment of the present invention is described in more detail below and in conjunction with the drawings, in which:

- Figure 1 shows a flowchart relating to the generation of a basic system;
- Figure 2 shows a flowchart relating to the generation of a customer-specific software system;

- Figure 3 shows a block diagram of an open drive regulator according to the invention; and
- Figure 4 shows a flowchart for the generation of the software for the open drive regulator shown in Figure 3.

#### DETAILED DESCRIPTION OF THE INVENTION

[0016] The software architecture on which the sequence in Figure 1 is based is broken down into the areas of the functions, alarms and parameters. In the example in Figure 1, the functions area contains a function block 1, a function block 2 and a function block 3. These are respectively associated with corresponding alarm blocks 1, 2 and 3, and parameter blocks 1, 2 and 3. To generate executable basic system software, the corresponding function, alarm and parameter blocks are joined together in the module 1. The module 1 then generates a corresponding executable code 2 for the basic system and lists 3.

[0017] Figure 2 shows the generation of a customer-specific system based on the basic system in Figure 1. In addition to the function, alarm and parameter blocks of the basic system, the system in Figure 2 contains OEM1 and OEM2 function, alarm and parameter blocks in the corresponding areas of the software architecture. The function block 2, the alarm block 2 and the parameter block 2 of the basic system are deselected by means of a user input. The functions of the function block 1, function block 3, OEM1 function block and OEM2 function block are then joined together with the corresponding alarm and parameter blocks in the module 1.

[0018] The process of the module 1 joining the corresponding blocks together then results in customer-specific code 4 and corresponding lists 5. This has the advantage that, depending on the customer-specific requirements, the customer can replace specific parts of the basic system by function, alarm and parameter blocks produced by the customer.

[0019] Figure 3 shows a block diagram of a corresponding drive regulator. The drive regulator has an object memory 6 for storing a system object 7 and function objects 8. The open drive regulator furthermore has a microcontroller 9, which is connected to the object memory 6, and also a user interface 10 for user-specific instantiation of one or more of the function objects 8. Furthermore, a memory 11 is provided for storing default values for the system object 7 and the function objects 8. The open drive regulator has a memory 12 for storing parameters, and a memory 13 for storing configuration data. For example, the open drive regulator can communicate via an interface 14 and a bus system, with a central unit or with other open drive regulators. Furthermore, it is also possible to introduce a smart card into the regulator via the interface 14 with user-specific data relating to the system configuration being stored on the smart card. The program for the open drive regulator is stored in the program memory 15.

[0020] When the open drive regulator shown in Figure 3 is started, the system object 7 is first of all loaded into the microcontroller 9. In this case, the system object 7 carries out the function of an operating system. If there are no customer-specific data items, then the system object 7 instantiates the function objects 8 of the basic system

using the default values in the memory 11 and joins them together to form an executable program, which is stored in the program memory 15 and the open drive regulator is started. Before and during the instantiation of the function objects, a user can enter user-specific data via the user interface 10, so that the instantiation of the corresponding function objects 8 can be carried out on a user-specific basis.

**[0021]** The user can likewise store user-specific function objects 8 in the object memory 6 corresponding to the OEM1 and the OEM2 function blocks in Figure 2. Corresponding user-specific inputs are stored in the memory 12 as parameters, and in the memory 13 as configuration data. When the open drive regulator is started once again, these parameter values are then accessed in order to instantiate the function objects 8 on a corresponding user-specific basis.

**[0022]** Figure 4 shows an embodiment of the method according to the present invention. First, the system object is loaded into the microcontroller in the step 40. Description data relating to the configuration of the system are read in step 41, in order to instantiate the system object. If necessary, the function objects are instantiated by means of user-specific parameters in step 42. In step 43, the corresponding parameters are stored. In step 44, the instantiated objects are compiled separately, and an executable program is generated from the corresponding compiled products. This program is started in step 45.